Open System Interconnection for Real-Time Factory Communications: Performance Results

Cornelis Franx
Philips
Eindhoven, The Netherlands

Kevin Mills
National Bureau of Standards
Gaithersburg, MD USA

This paper considers the applicability of Open System Interconnection (OSI) protocols, as now defined and implemented, for use in real-time factory communications. Factory communications requirements are described by outlining the hierarchial nature of the factory network architecture and by defining the nature of real-time at the lowest level of hierarchy. Two possible solutions for real-time factory communications are described: 1) the full MAP seven layer architecture and 2) the MAP enhanced performance architecture (EPA). Measured performance of a five layer OSI protocol implementation is described with special emphasis on one-way delays. Measurement results are also given for throughput. The ability of present OSI standards to guarantee real-time performance is evaluated. A flow control problem is identified concerning use of an OSI transport protocol over a type 1 class 1 logical link control protocol.

I. Introduction

Data communications within a factory must meet a hierarchy of performance requirements including real-time at the lowest levels and time-critical within a workcell. The Manufacturing Automation Protocol (MAP) standard, requiring all seven OSI protocol layers, was thought inadequate for real-time and time critical applications. Thus, a subset of protocols defining an Enhanced Performance Architecture (EPA) has been added to the MAP standard. The purpose of this paper is threefold: 1) to describe the hierarchical performance requirements within a typical automated factory, 2) to explain the differences between the full seven layer MAP standard and the three layer EPA, and 3) to provide measured performance results for applications using the lower four layers of the MAP standard.* Each of these topics is covered in a separate section below. Some conclusions are drawn with respect to the performance possibilities of EPA and the full MAP protocols and areas of further research are indicated.

II. Factory Communications Requirements

Communication requirements in a factory depend on the type of production process carried on in the factory. So the production process and the related production control architecture have to be described before the communication requirements can be stated.

In Philips there are many factories with discrete assembly lines, for instance to produce radio sets. The control structure of these lines will be based on the National Bureau of Standards hierarchical model for production control that was adopted by Philips (see Figure II-1) [ALB81]. The controller processes at different levels in the hierarchy will be implemented on separate computer systems. In the automation module and device control layers individual systems control robots, positioners and local workpiece transport within an assembly station. The workstation controller sequences the activities of the automation module controllers to execute assembly tasks with elapsed time in the two to five second range. The workcell controller routes incoming parts to available stations, coordinating the area transport and the assembly workstations. Data communication between the computer systems is required when their controller processes interact.

---

*Certain commercial equipment is identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the equipment identified is necessarily the best available for the purpose.

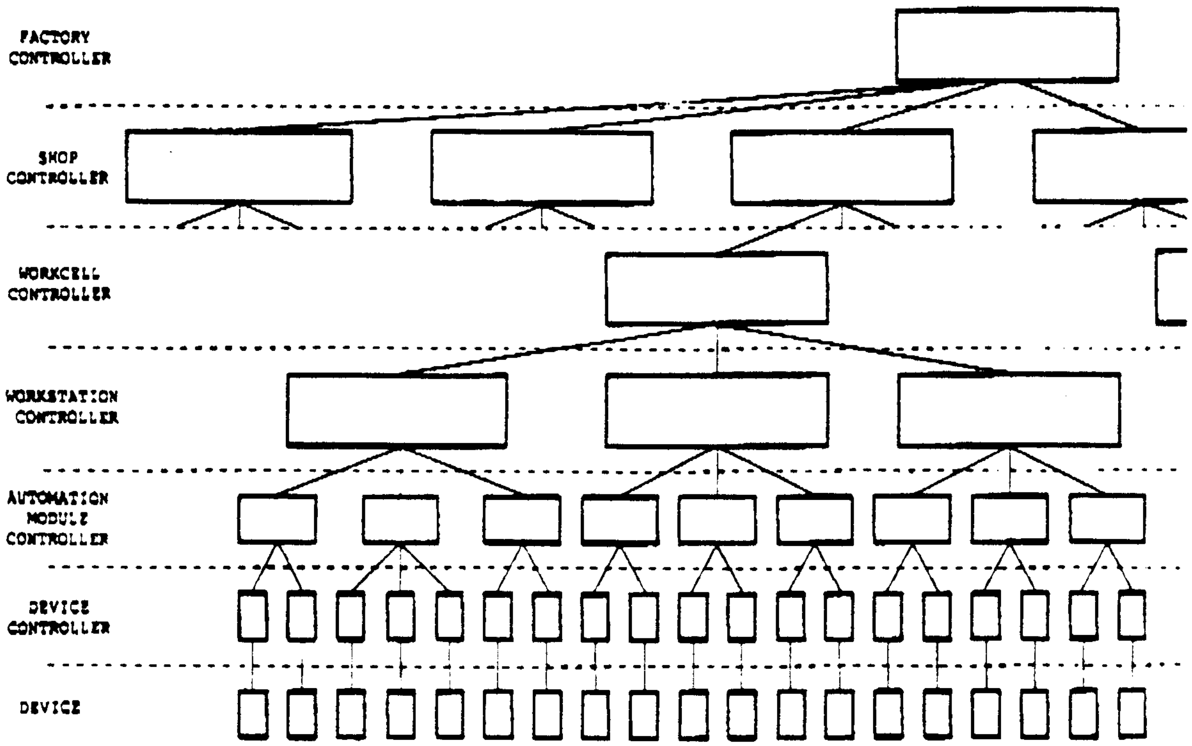

FIGURE II-1. LOGICAL MODEL OF A PRODUCTION CONTROL HIERARCHY

The time requirements of a controller process become less stringent as the process is positioned higher in the model. Controllers at the top of the model are concerned with long term planning and control, at the bottom of the model they do direct real-time control. At the upper layers large batches of data (> 1Mb) have to be transported at one time, for instance a daily production plan. This batch data must be transported within several minutes. Going down in the model time requirement become shorter and more stringent, but the amount of data to be transported at one time decreases as well.

In the Philips environment, from the workcell controller down, communication times on the order of one second can reduce production throughput by 10-20%. Consider, for example, the communication between workcell controller and workstation controller. At this level work orders and status reports are exchanged with message lengths of 100 - 500 bytes. A message must be sent and delivered within 100 to 200 ms.

Communication at the next lower level between automation module controllers and workstation controller is more critical, with 100 ms being a maximum time for communications. Communication within the workstation becomes highly time critical, especially between device controllers and automation modules. Control loops at this level require small amounts of data (< 20 bytes) to be transmitted very fast (within 10 ms) with a repetition frequency of up to once every 10 ms. All figures presented here are meant to give a global indication of the communication requirements and they only concern the delivery of production control information.

When there are tight time limits like those at the bottom level, the traffic is often called "real-time" generally without defining the meaning of "real-time". In this paper it has the following meaning:

> Communication is real-time when a message must be passed from one process to another within a previously specified time limit in order for the process to correctly perform its function.

The data communication network has to guarantee that except in the case of system or component failure the message will be transferred within that time limit. To prevent messages with a more relaxed time limit from interfering with urgent messages and delaying them, priorities have to be allocated to messages. Higher priority messages are handled before messages with a lower priority.

Another important aspect of real-time communication is its error behavior. In case of a link failure only limited time should be spent on error recovery procedures, then the application processes must be warned to enable them to take appropriate action.

As mentioned eariler, real-time communication can be found at the bottom of the model, however, it is important to realize that not all communication is for the purpose of production control. Communication for software downloading and reporting of production statistics has different requirements. Generally the time limits are not very tight in these cases but reliability is more important, requiring more extensive error recovery efforts.

The different communication requirements at the bottom two layers of the factory model demonstrate the need for a network that offers a real-time communication service as well as a more reliable, higher level communication service. Without a standardized network offering both types of service, proprietary networking solutions are inevitable.

III. OSI and MAP Enhanced Performance Architecture

Two possible architectures for real-time factory communications will be discussed, the "full" MAP architecture based on the seven layer OSI reference model and the three layer Enhanced Performance Architecture (EPA). In MAP version 2.1 a selection of protocols and options for six layers of the OSI model (Fig. II-2) has been made. The missing protocol for the presentation layer will be supplied later. Based on this selection of protocols, vendors have started to make interoperable data communication products for the factory.

Almost from the beginning of MAP there were doubts from the process control industry that seven layer MAP (called "full" MAP) could meet the performance requirements for real-time applications. This led to the introduction of a new, three layer architecture called the enhanced performance architecture (EPA) (Figure II-3). Many of the ideas behind EPA were adopted from Proway which is a local area network for the industrial environment defined by ISA SP 72 [ISA72].

Both full MAP and EPA use at their bottom layers the IEEE 802.4 token bus [IEE85]. The first important difference between the two is at the data link layer where EPA uses logical link control (LLC) type 3, acknowledged connectionless service, instead of LLC type 1, unacknowledged connectionless service [IEE84]. LLC type 3 confirms the arrival of data at the destination LLC layer and retransmits data in case of errors. More precisely when a station has the token it sends an LLC type 3 frame and then waits for a returning LLC type 3 frame that is in fact an acknowledgement. Without this response, a timeout will occur and the LLC frame will be retransmitted. Only after the acknowledgement is received or the maximum number of retransmissions is exhausted can the token be passed to the next station. It is notable that this LLC protocol interacts with the operation of the token passing protocol at the MAC layer.

The time the station can hold the token (token-hold time) is limited; therefore, the LLC type 3 acknowledgement and retransmission scheme has to work very fast, and only three retransmissions are allowed. Without a tight limit on the token-hold time the total performance of the token bus could degrade severely when there are repeated LLC type 3 failures.

The LLC type 3 timing characteristics make it impossible to transit a bridge because a bridge introduces unacceptably large message delays when it passes LLC frames from one network segment to another. As a result the use of LLC type 3 is limited to a single segment.

LAYERS	MAP V2.1 PROTOCOLS	MAP/EPA PROTOCOLS
Layer 7 Application	ISO FTAM (DP) 8571 File Transfer Protocol, Manufacturing Message Format Standard (MMFS) and Common Application Service Elements (CASE)	EIA MMS RS 511 Manufacturing Message Service
Layer 6 Presentation	NULL (ASCII and Binary Encoding)	NONE
Layer 5 Session	ISO Session (IS) 8327 Basic Combined Subset and Session Kernel, Full Duplex	NONE
Layer 4 Transport	ISO Transport (IS) 8073 Class 4	NONE
Layer 3 Network	ISO Internet (DIS) 8473 Connectionless Network Service (CLNS)	NONE
Layer 2 Data Link	ISO Logical Link Control (DIS) 8802/2 (IEEE 802.2) Class 1, Type 1	Class 3, Type 3
Layer 1 Physical	ISO Token Passing Bus (DIS) 8802/4 (IEEE 802.4) Broadband	Carrierband

FIGURE II-2. MAP V2.1 AND FUTURE MAP/EPA PROTOCOL STACK

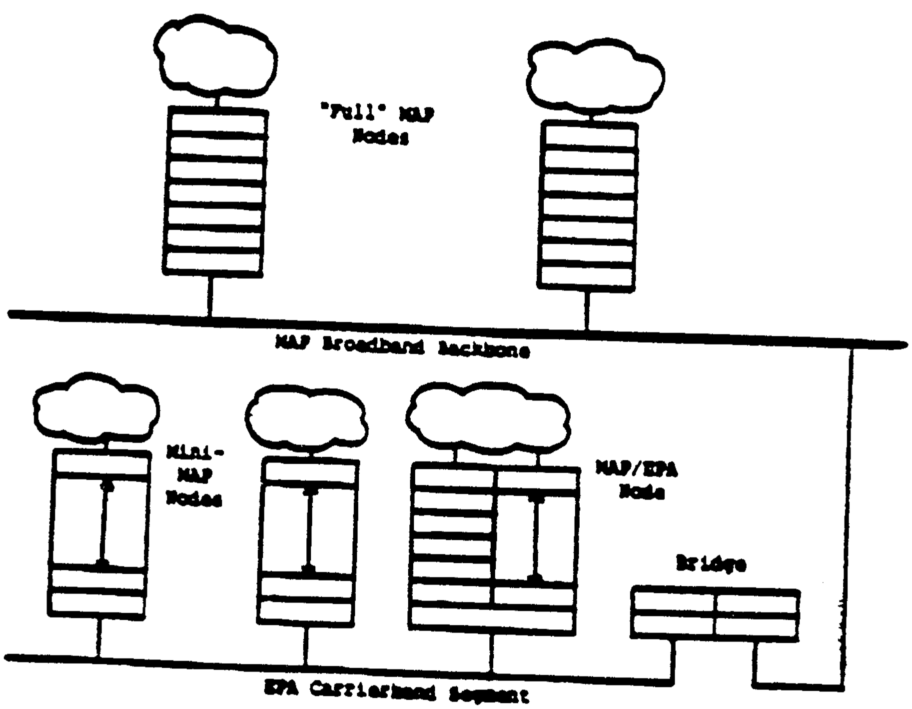

FIGURE II-3. NODES ON AN EPA SEGMENT

The use of LLC 3 to provide real-time message delivery is not without potential problems. When a token-holding station is awaiting an immediate acknowledgement several events can occur. For example, the token-holding station may retransmit the message or the token-holding station may give up and pass the token. Either of these messages may collide with the acknowledgement. Collision with the retransmitted message will cause another retransmission. Collision with the token will invoke token recovery procedures.

It is very likely that EPA, in addition to the bottom two layers with IEEE 802.4 token bus and LLC type 3, will ultimately have a third protocol layer containing Manufacturing Message Service (MMS) which runs directly on top of the LLC. The mapping between MMS and LLC has already been described in a document prepared by MAP's Programmable Devices Committee.

In the EPA specification layers 3 - 6 of the OSI model are absent. Because of the missing layers EPA offers reduced functionality when compared to "full" MAP. The consequences of skipping four layers will be discussed briefly, starting at the network layer.

Because there is no network layer it is impossible to traverse intermediate systems (routers). But, since intermediate systems introduce relatively large and unpredictable message delays, this limitation is necessary to achieve real-time performance. Without a network layer it is possible to leave out the integrity assurance protocol of the transport layer as well because LLC type 3 already offers end-to-end acknowledgment. Some of the missing transport functions such as, resequencing, flow control, and multiple associations, are covered by the mapping between LLC type 3 and MMS.

The next higher layer, the session layer, offers dialogue control and resynchronization which are not suited for real-time applications. Therefore, the session layer can be skipped as well. Finally EPA has a presentation syntax (X.409) [CCI84] and an application layer protocol when the layer seven protocol, MMS [EIA85], is added to EPA.

The use of EPA and "full" MAP can be combined on a single EPA segment (Figure III-3). On the segment there can be MAP/EPA nodes with the three and seven layer stack and Mini-MAP nodes with the three layer stack only. The choice between "full" MAP or EPA should be based on the communication requirements of the applications. Applications in a MAP/EPA node select the seven layer stack to get the functionality of "full" MAP and then are able to communicate with all other MAP/EPA nodes on the segment and with "full" MAP nodes on the broadband backbone. Real-time applications select the EPA stack to get faster responses and then are able to communicate with other MAP/EPA nodes and Mini-MAP nodes on the same segment. Real-time applications must be aware of the reduced functionality of EPA.

The main differences between "full" MAP and EPA have been pointed out. Now the suitability of both architectures to handle real-time communication will be discussed. The functionality of "full" MAP is not always required, and the associated protocol activities are sometimes undesirable for real-time communication (i.e., message routing, extensive retransmissions and resynchronization). When the seven layer OSI stack is used, message delays are unpredictable and can not be controlled. One of the main reasons for this is the lack of control over the allocation of system resources making it impossible to give real-time communication priority over other communication and activities like network management (see Section IV). Without priorities all communication is treated in the same way regardless of its urgency and it is possible that large file transfers will interfere with short real-time messages and delay them.

EPA is in a better position to provide real-time communication than "full" MAP. Because it has only three protocol layers, processing delays can be much smaller. Moreover, it has a priority mechanism at the LLC layer that, when MMS honors priorities as well, permits full application-controlled use of priorities. Message delays are predictable when the number of nodes in the segment is known and the segment is in a stable operating condition. It should be realized that this is only true for messges with the highest priority -- lower priority messages are delayed by higher priority messages making it more difficult to predict their delay. LLC type 3 has a favorable error recovery mechanism because it performs retransmissions in a short time and then warns the application. Other failures delaying the transmission of a message should be reported as well (i.e., the collapse of the token ring).

For real-time communications the EPA stack seems to have attractive properties, but Philips is concerned about the cost to produce error-free application software for EPA and about the portability and flexibility of this software. Also, conformance and interoperability testing will be difficult for EPA because there are many options, such as connectionless or connection-oriented use of MMS, MMS subsets, and non-token stations.

IV. OSI Measured Performance

In order to assess the suitability of OSI protocol standards for factory applications, the National Bureau of Standards (NBS) and Philips conducted a cooperative project to measure the communication performance in a small testbed network using OSI protocol implementations. Intel Corporation donated hardware and software for the project. A brief description of the testbed network is given, followed by a discussion of the performance results.

A. The Testbed Network

The local area network testbed implemented at the NBS is illustrated in Figure IV-1. Four Intel 310 nodes and a passive, real-time monitor are connected to a CSMA/CD local network.* A global clock circuit is connected to each Intel 310 node to provide a synchronized measurement clock. The internal architecture of each node is shown in Figure IV-2.

OSI communication services are provided by Intel's iNA-960 software [INT84A] running on a 186/51 COMMputer TM board [INT84B]. The 186/51 contains two processing elements: an 80186 (transport, network, and logical link control) and an 82586 (media access control). Traffic generation and measurement software runs on a separate host board based on an 80286 processor [INT83]. Communication between the host and COMMputer board is via message passing using the Multibus Interprocessor Protocol (MIP) [INT84A, Appendix E].

Figure IV-3 shows how the global clock board provides a synchronized 100 us pulse to each Intel 80286 CPU board. The clock pulse is connected to a 16-bit programmable interval timer (PIT). The PIT overflows every 6.5 seconds causing a 16-bit software clock to be updated. The entire 32-bit clock is available to user software.

Figure IV-4 illustrates the time delay measurements made within the measurement software. A user task requests communications services by issuing a request block (RB) to iNA-960 via a system call. The time required to return from the system call is measured as T1. Once the iNA-960 has provided a requested service, the RB is returned to the user program. T3 measures the time elapsed between issuance of the RB and its return. An RB normally contains a user message within it. T2 measures one-way delay for user messages. T4 measures the duration of an experiment.

The variables controlled by the traffic generation software are listed below (Table IV-1). Another set of variables, such as retransmission timer values and transport message sizes, are controlled on a connection-by-connection basis using iNA-960 network management services. The network management services are also used to monitor lower level measures such as collision counts, count of packets sent and received, and number of packets dropped due to buffer overrun. A passive, real-time monitor enables unobtrusive evaluation of experiment progress -- indicating number of connections, number of retransmissions, protocol efficiency, and total data sent [MIL85].

The experiments divide naturally into three sets: 1) throughput profile, 2) delay profile, and 3) multi-application profile. Measured results for each set are discussed in the following sections.

----

*Although MAP requires use of a token passing bus media access control technique, the object of this study is transport layer performance on an unloaded local network, and so the use of a CSMA/CD local network does not invalidate the study.

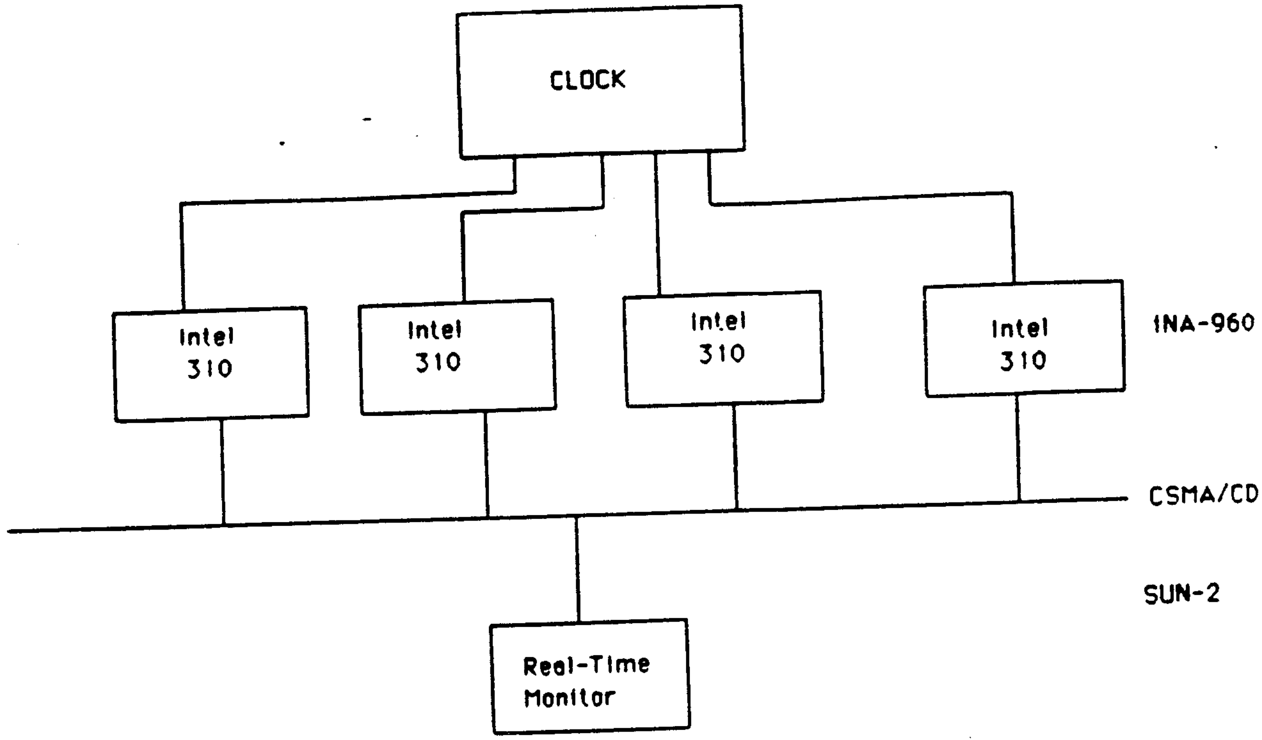

FIGURE IV-1. EXPERIMENT NETWORK TESTBED

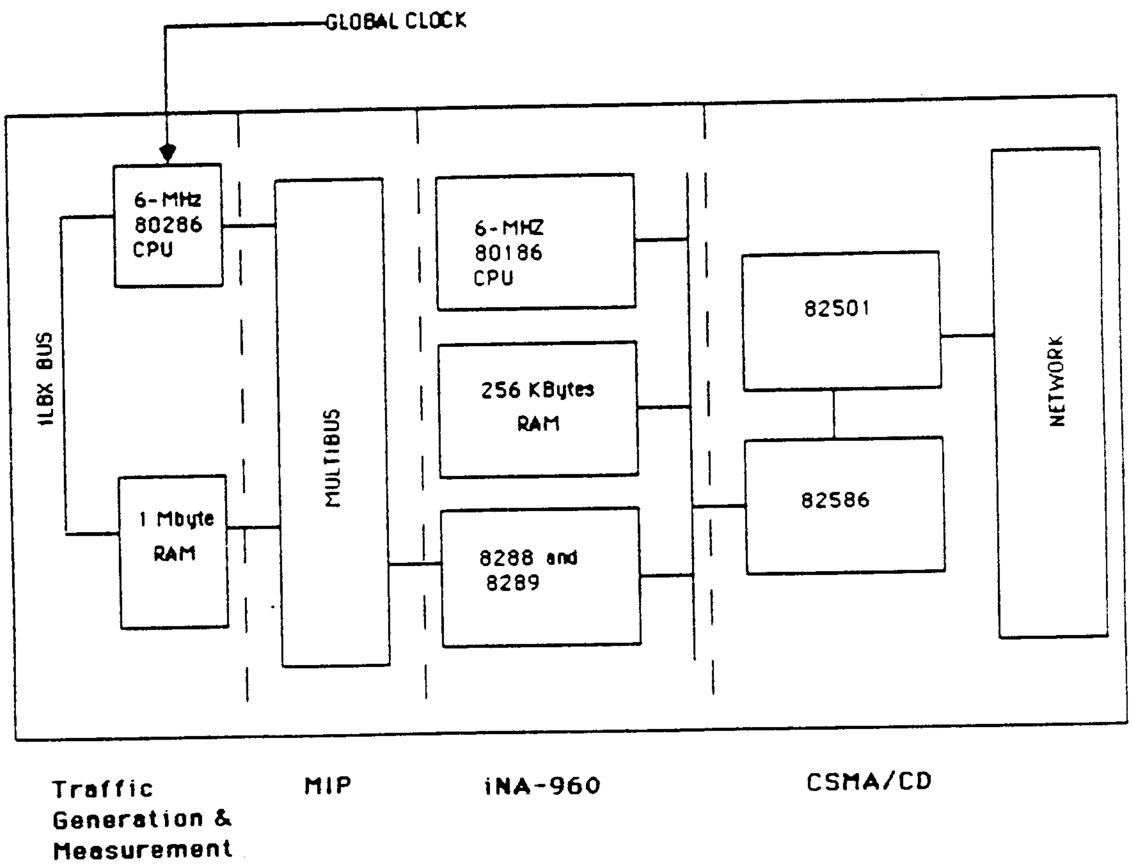

Figure IV-2. Architecture of an Intel 310 Network Node

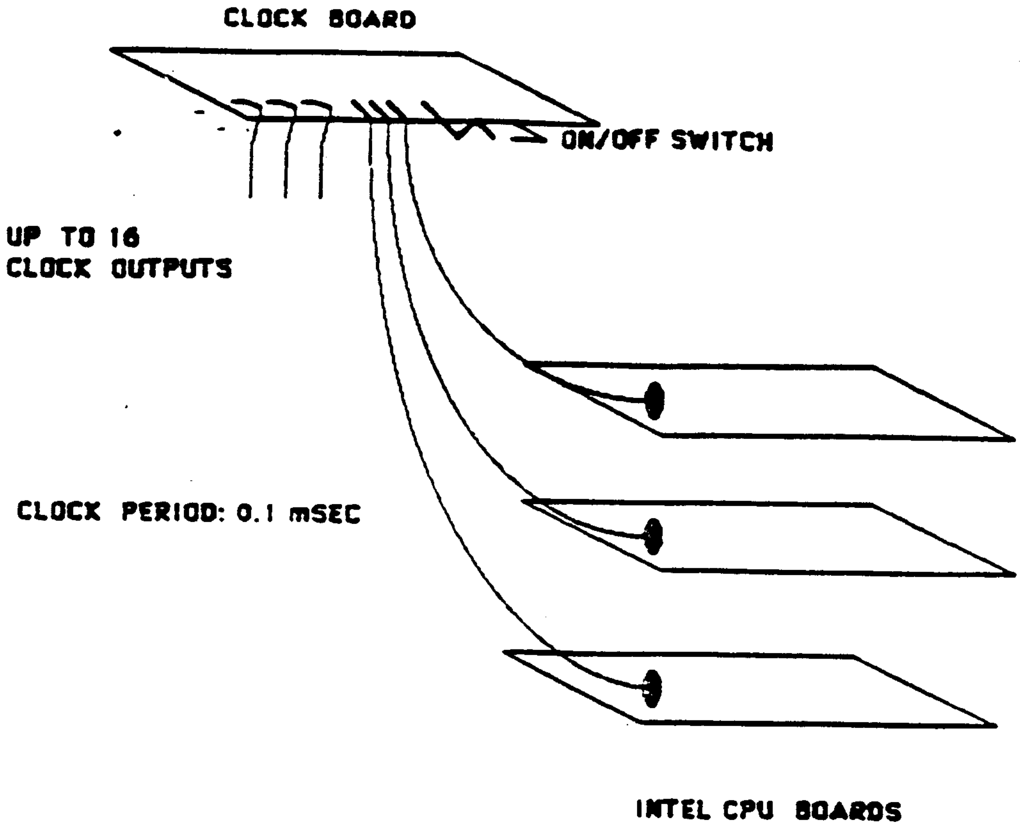

FIGURE IV-3. THE GLOBAL SYNCHRONIZED CLOCK SYSTEM

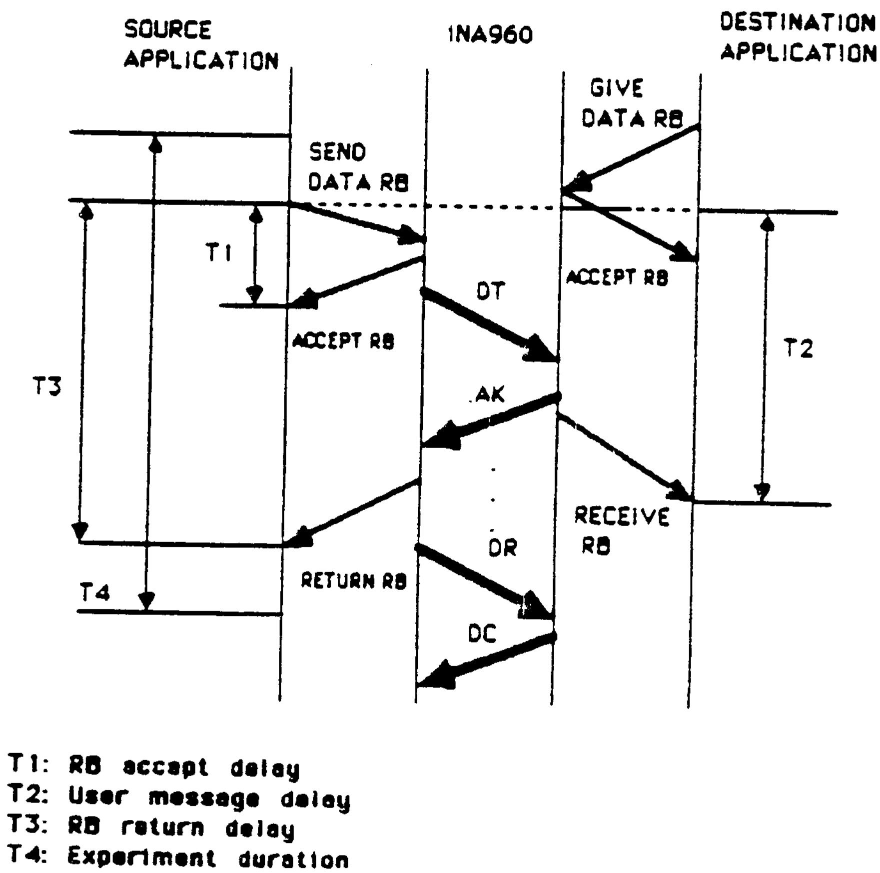

FIGURE IV-4. USER DELAYS MEASURED

Table IV-1. <u>Traffic Generation Variables</u>

Application Priority
Inter-message Delay
Duplex or Simplex Data Flow
User Message Size
Total Data Transferred
Number of Connections
Number of User Buffers

B. Throughput Profile

The throughput profile shows total measured throughput under a variety of conditions. Simplex and full-duplex data flows are considered and a flow control problem is discussed. User message sizes are always 10K octets. To achieve the best throughput the value for the adaptable retransmission timer parameters had to be increased as the number of connections increased at each node. This adjustment is required because the apparent round-trip time increases as the load increases in each node.

Simplex Transfer

Figure IV-5 shows the total throughput measured during simplex data transfer, between two Intel 310 systems, as the number of buffers per connection is varied. Measures are shown for one, two, three and four transport connections. The minimum values for the adaptable retransmission timers used for each experiment are given below (Table IV-2). Throughput ranged between 60 Kcps and 108 Kcps. With only two buffers per connection end-point available, throughput is increased (Figure IV-5) by adding connections because unused CPU capacity is available within the system. Once four buffers are available per connection, the unused capacity is reduced and the overhead associated with connection scheduling becomes evident. Little throughput difference was observed between three and four connections.

Full-Duplex Transfer

Figure IV-6 shows throughput measured when the experiments were repeated using full-duplex data transfer. Retransmission timer values used are shown in Table IV-3. Throughput ranged between 90 Kcps and 104 Kcps.

Flow Control Problem

During the throughput experiments a problem was found with the combination of the OSI transport protocol operating over a type 1 class 1 logical link control protocol. The problem is illustrated using the two throughput curves shown in Figure IV-7. One curve (single sender) shows a pair of identical machines engaged in a two-connection simplex data transfer. As the number of transport buffers per connection increases, the throughput increases toward 104 Kcps. No matter how many transport receiver buffers are offered, the receiver's link buffers cannot be overrun because only two machines are involved and both machines have identical processing capabilities.

The second curve on Figure IV-7 (two senders), indicates what happens if the sending machine is faster than the receiver or if two machines are sending to one receiver. As the number of transport receive buffers per connection increases, the throughput decreases toward 35 Kcps. The lost throughput occurs because the transport flow control window, a direct reflection of the number of receive buffers, allows the link level buffers of the receiver to be overrun, invoking transport layer retransmission procedures. Use of link layer flow control would solve this problem; otherwise, transport layer buffer decisions must be made by accounting for link layer buffer conditions.

Table IV-2. Retransmission Timer Values for Simplex Throughput

Connections	Minimum(Secs.)	Starting(Secs.)
1	.256	.512
2	.512	1.024
3	.819	1.638
4	1.024	2.048

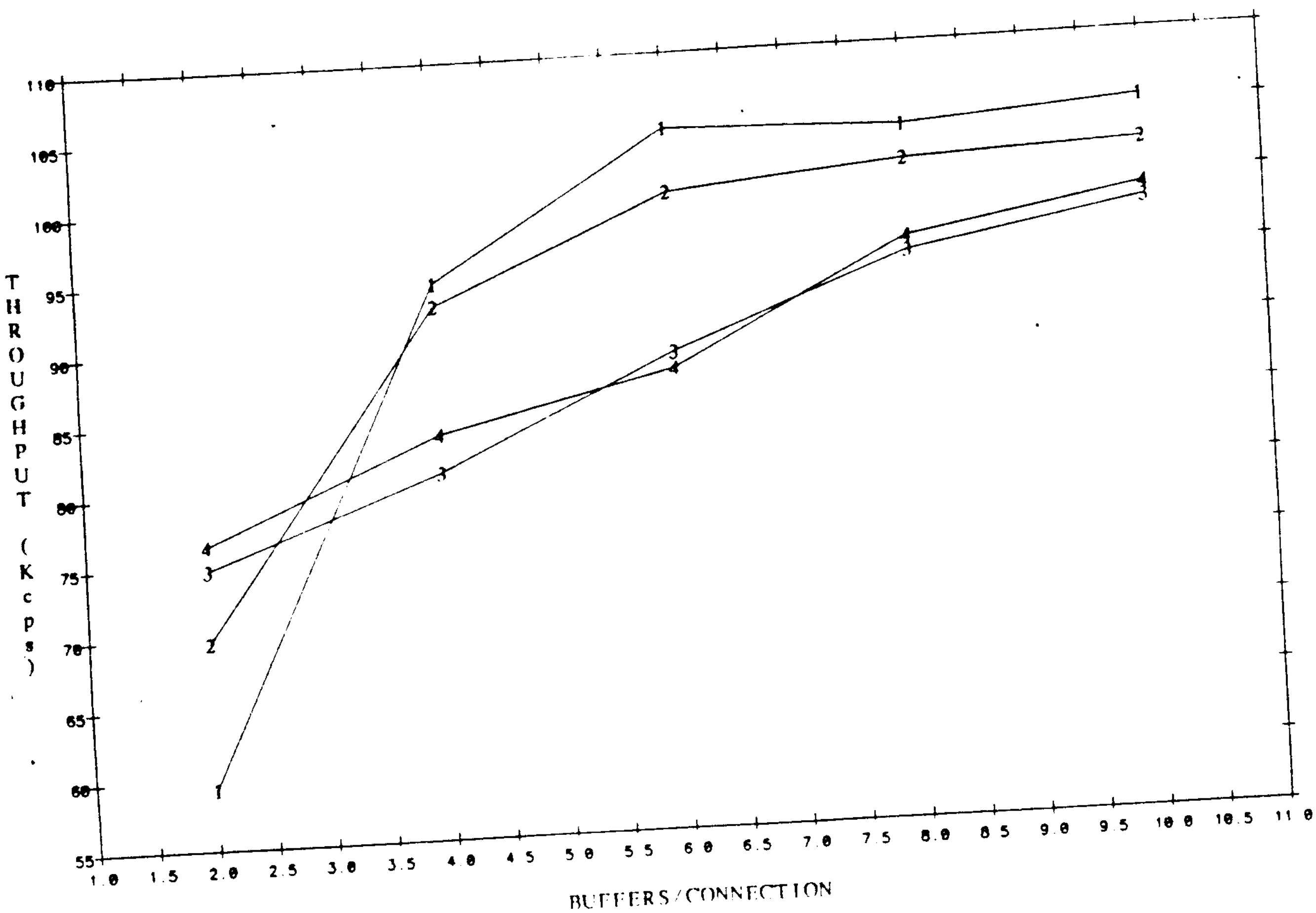

FIGURE IV-5. SIMPLEX THROUGHPUT PROFILE

Table IV-3. Retransmission Timer Values for Full-Duplex Throughput

Connections	Minimum (Secs.)	Starting (Secs.)
1	.512	1.024
2	1.024	2.048
3	1.638	3.276
4	2.048	4.096

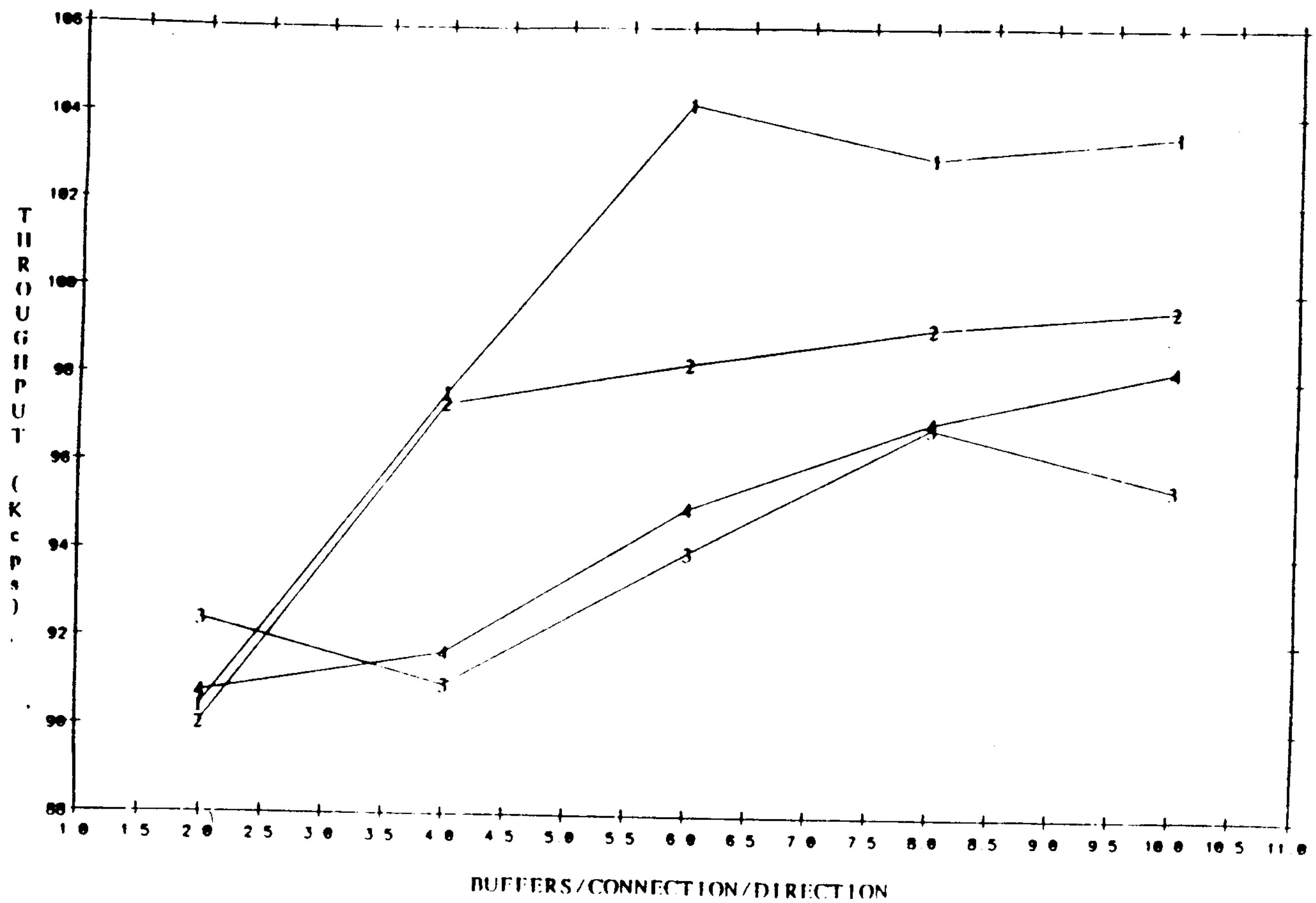

FIGURE IV-6. DUPLEX THROUGHPUT PROFILE

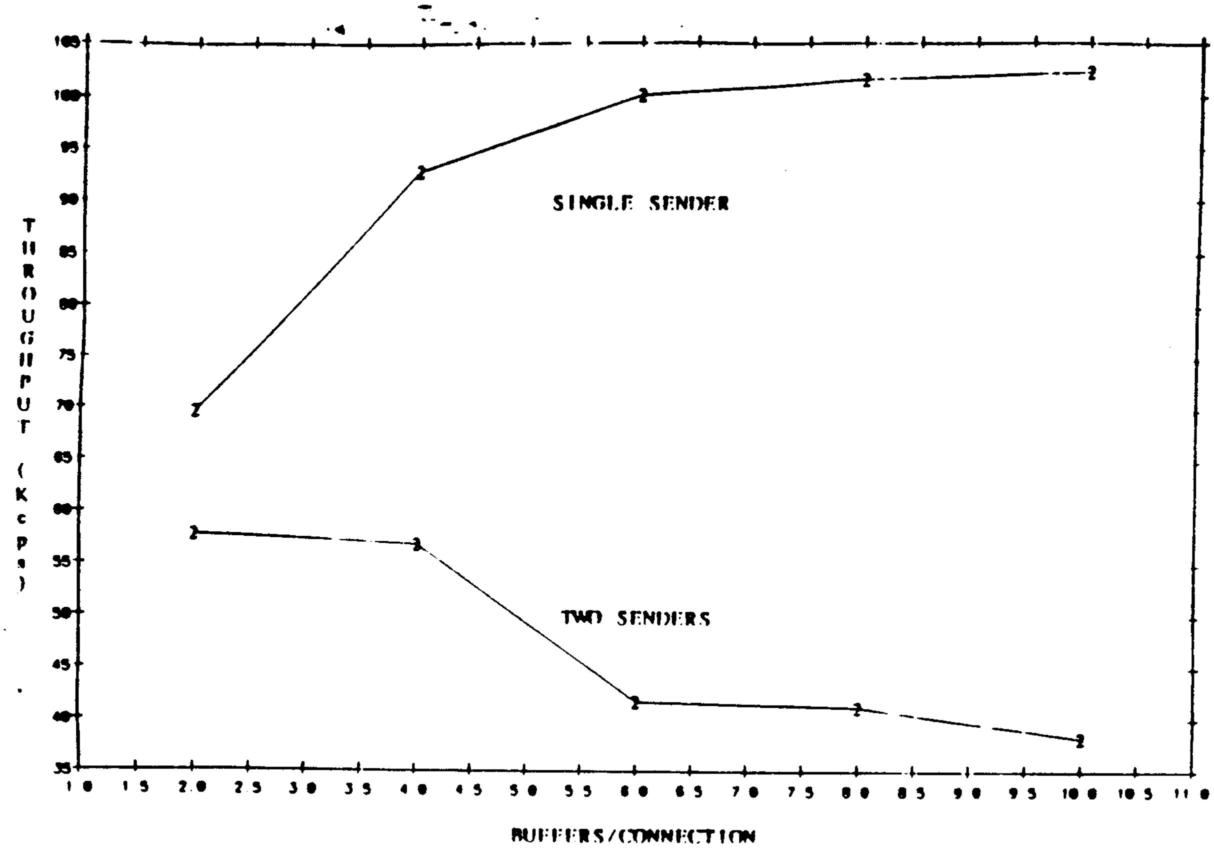

FIGURE IV-7. RECEIVER OVERRUN

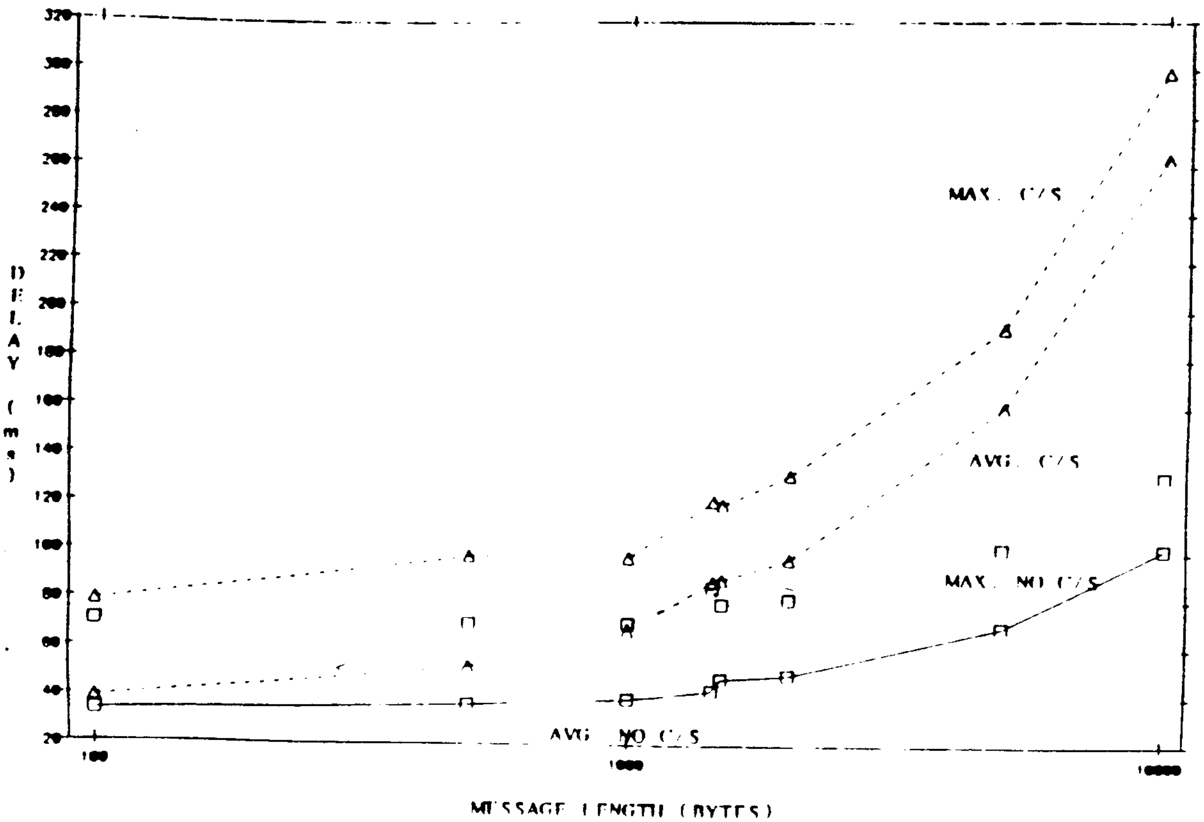

FIGURE IV-8. ONE-WAY DELAY PROFILE

C. One-Way Delay Profile

This section presents a profile of one-way user message delays (T2 in Figure IV-4) measured under a variety of conditions. In all of the delay experiments, the user message size is varied between 100 and 10,000 octets. However, when a user message is large, protocol segmenting is required because each link packet will hold only 1500 data octets. The sending user on each connection submits one message and waits for an acknowledgement indication before submitting the next message. This stop-and-wait operation limits the overall load on iNA-960 during the delay experiments.

Single Connection Delays

Figure IV-8 presents measured one-way delays with and without checksum enabled. The message transfers occur over a single connection in a single direction. The receiver allocates three transport receive buffers so that no delay is incurred for closing and reopening the transport flow control window. The lowest delays obtained occur with 100-octet user messages and no checksum, 33.5 ms average and 70.5 ms maximum. The addition of the checksum raises the lowest delays to 38.4 ms average and 78.8 ms maximum. As expected, the effect of the checksum on delay is more significant as the message size increases.

Multi-connection Delays

Figures IV-9 and IV-10 illustrate the effect of multi-connection traffic on one-way delays. For the results in Figure IV-9 the receive buffers are limited to one per connection. Thus, the effect of closing and reopening the transport flow control window is evident. The lowest one-way delays are obtained with a single connection and 100-octet messages, 45.3 ms average and 88.6 ms maximum. This means that, on the average, 11.8 ms is required to handle reopening of the transport flow control window (comparing Figure IV-9 with Figure IV-10). In the maximum case, 18.1 ms is required. While this overhead increases delay at small message sizes, it serves to reduce the one-way delay as the message size increases. Forcing the extra delay to reopen the window on each connection reduces the increase in traffic intensity normally associated with larger user message sizes.

Figure IV-10 illustrates the same experiment with three transport receive buffers allocated to each connection. As the load increases for two, three, and four connections, the average and maximum one-way delays increase significantly. The upper bounds on one-way delay in the previous case were 443.2 ms average and 904.5 ms maximum. The upper bounds in this experiment are 1395.3 ms average and 2076.0 ms maximum.

An increase of this magnitude is almost certainly due to a queuing delay incurred at the receiving user program. The user program submits empty receive buffers to and accepts filled received buffers from iNA-960. As configured, iNA-960 gives a higher priority to processing of transport operations, including passing filled receive buffers to the user, than to accepting empty receive buffers from the user. Therefore, a user's receive queue grows during periods when the user program is blocked waiting for iNA-960 to accept an empty receive buffer. This effect is demonstrated by an

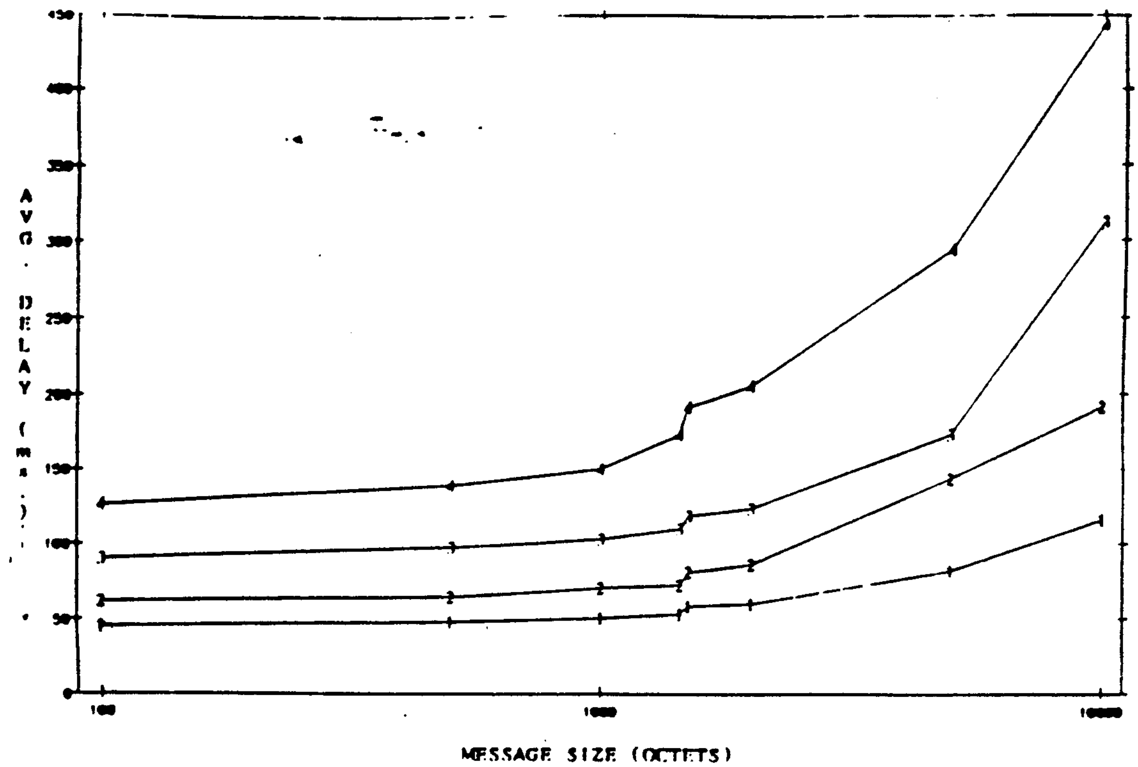

FIGURE IV-9. MULTI-CONNECTION DELAY PROFILE WITH TAUT FLOW CONTROL

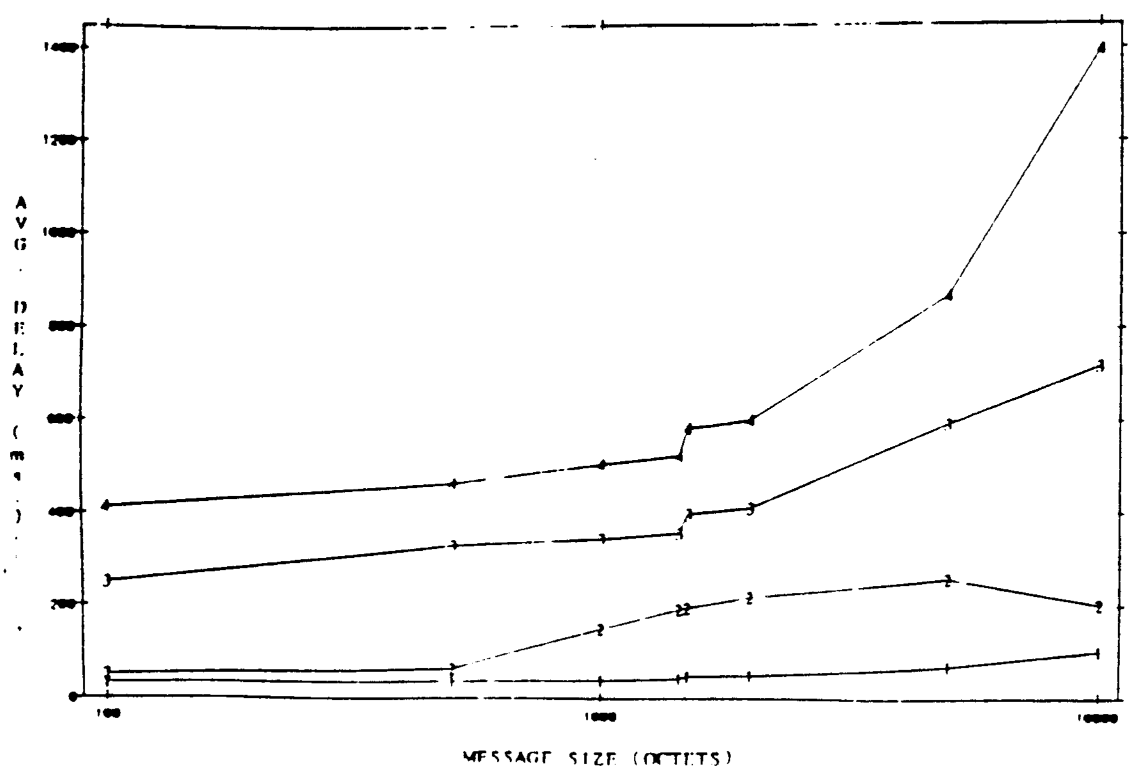

FIGURE IV-10. MULTI-CONNECTION DELAY PROFILE WITHOUT TAUT FLOW CONTROL increasing request block accept delay (T1 in Figure IV-4) as iNA-960 traffic intensity increases. This effect might be reduced if the MIP task on the 186/51 is run at a higher priority than the iNA-960 task.

D. Multi-Application Profile

The next set of experiments involves a pair of traffic generation tasks on each of two Intel 310 systems. The first pair of tasks is generating bulk data traffic. The second pair of tasks simulates a status report application, submitting messages at a rate sustainable by the system so that no queuing delay is included. The load caused by the bulk data transfer is controlled by varying, between 400 octets and 40K octets, the size of transmit and receive buffers. Status report messages are fixed at 100 octets. The operating system priority of the status reporting task is higher than that of the bulk data task.

Figure IV-11 shows the experiment results when the data flow for both connections is in the same direction. The abscissa plots throughput of the bulk data transfer. The ordinate plots the average one-way delay for status report messages. Ideally the status report message delays (average and maximum) will be kept near the lowest delays available from the system. These target delays are superimposed on the graph with dashed lines.

The results show that the status report delays increase in a pattern similar to that seen when message sizes increase (Figure IV-10, two connections) though the status report messages do not increase in size. Also note that the lowest average and maximum delays are twice the ideal. These results represent unacceptable behavior for applications requiring real-time response. The only control mechanism available in the OSI transport standard is the allocation of buffer space. Therefore, iNA-960 does not provide priority scheduling for transport layer connections and the MIP implementation contains no multi-queue mechanism. Thus, the operating system task priority is not complemented by necessary control mechanisms in the communication system.

Figure IV-12 gives the results of the same multi-application experiment except that status reports and bulk data flow in opposite directions. These results show the lowest average delay is five times the ideal, while the lowest maximum delay is three times the ideal. Although these results are much worse than for the simplex case, the delays do not rise much as the bulk data throughput increases.

V. Conclusions

OSI protocol standards, as now specified, do not provide adequate mechanisms for guaranteeing real-time performance for selected connections or messages. This weakness in the standards is demonstrated by the iNA-960 multi-application profile where high throughput transport connections dominate the available resources forcing up the delay on all connections.

The performance measurements reported suggest limits to the traffic that can be served by first generation OSI protocol implementations. For iNA-960, a typical OSI transport layer product, an upper bound on throughput of 108 Kcps

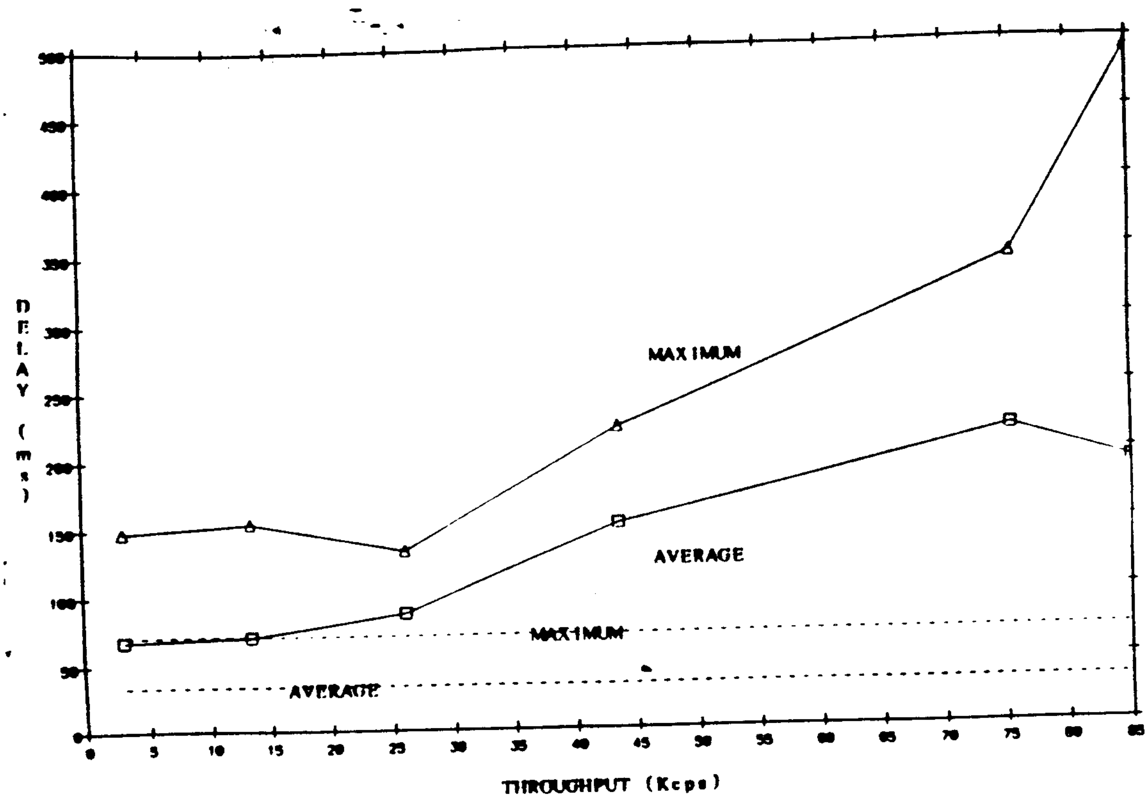

FIGURE IV-11. DUAL-APPLICATION (SIMPLEX)

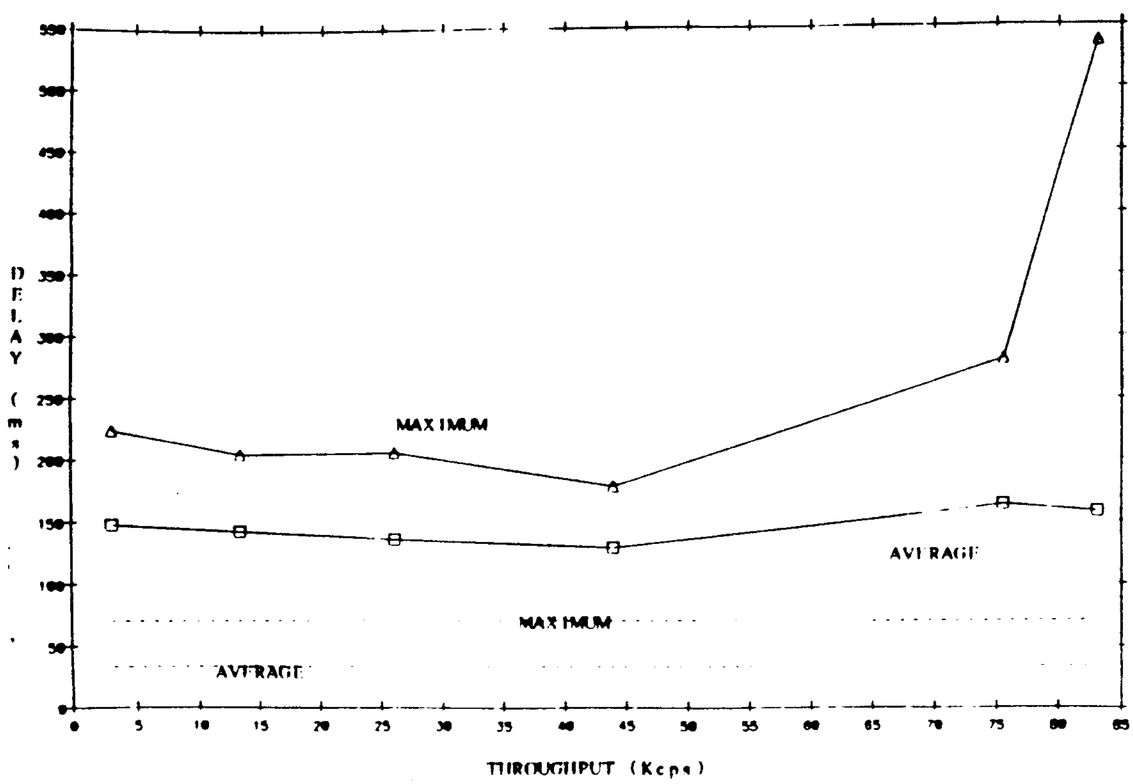

FIGURE IV-12. DUAL-APPLICATION (DUPLEX)

and a lower bound on delay of 33.5 ms were measured. The measured performance decreased as the number of connections increased. A flow control problem may occur when an OSI transport protocol is used over a type 1 class 1 logical link control protocol.

As long as "full" MAP cannot meet the complete hierarchy of performance requirements for factory communications solutions like EPA are inevitable. The EPA, with fewer protocol layers and a potential for application access to priority queues, has better properties for achieving real-time communication. Although the real-time performance of EPA is expected to be much better than with "full" MAP, this has not been demonstrated. The LLC 3, with inherent conflicts between token passing and contention, may prove difficult to implement successfully and may not provide, deterministically, fast data transfer between devices from multiple vendors. MMS may possibly turn out to be an important performance bottleneck. The performance of EPA needs to be investigated in a manner similar to that reported here for OSI protocols.